This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) An apparatus for producing a new ((n)th) black box for a digital rights management (DRM) system, the (n)th black box for being installed in the DRM system and for performing decryption and encryption functions in the DRM system, the (n)th black box being produced and delivered to the DRM system upon request therefrom and including a new ((n)th) executable and a new ((n)th) key file, the (n)th key file having a new ((n)th) set of black box keys and a number of old sets of black box keys, the request including an old ((n-1)th) key file having the old sets of black box keys, the apparatus comprising:

a code optimizer / randomizer receiving a master executable and randomized optimization parameters as inputs and producing the (n)th executable as an output; and

a key manager receiving the (n-1)th key file and the (n)th set of black box keys as input, extracting the old sets of black box keys from the (n-1)th key file, and producing the (n)th key file including the (n)th set of black box keys and the old sets of black box keys as an output;

wherein the (n)th executable and the (n)th key file are to be forwarded to the requesting DRM system,

the key manager producing the (n)th key file encrypted according to a secret, the apparatus further comprising an injector receiving the (n)th executable from the code optimizer / randomizer as an input, injecting the secret into the (n)th executable in a pre-determined location, and producing the injected (n)th executable as an output, wherein the

injected (n)th executable and the encrypted (n)th key file are to be forwarded to the requesting DRM system.

2. (Canceled)

3. (Currently Amended) The apparatus of claim [[2]] 1 wherein the key manager produces the (n)th key file encrypted according to a symmetric key, the apparatus comprising an injector receiving the (n)th executable from the code optimizer / randomizer as an input, injecting the symmetric key into the (n)th executable in a pre-determined location, and producing the injected (n)th executable as an output, wherein the injected (n)th executable and the encrypted (n)th key file are to be forwarded to the requesting DRM system.

4. (Currently Amended) The apparatus of claim [[2]] 1 wherein the (n)th set of black box keys includes a public – private key pair, and wherein the key manager produces the (n)th key file encrypted according to the private key, the apparatus comprising an injector receiving the (n)th executable from the code optimizer / randomizer as an input, injecting the private key into the (n)th executable in a pre-determined location, and producing the injected (n)th executable as an output, wherein the injected (n)th executable and the encrypted (n)th key file are to be forwarded to the requesting DRM system.

5. (Currently Amended) The apparatus of claim [[2]] 1 wherein the injector injects the secret into the (n)th executable in the pre-determined location such that the secret is hidden except to the (n)th executable.

6. (Currently Amended) The apparatus of claim [[2]] 1 wherein the DRM system resides on a computing device has a hardware ID (HWID) associated therewith, wherein the HWID is included in and obtained from the (n-1)th key file, and wherein the injector injects the obtained HWID into the (n)th executable in a pre-determined location.

7. (Currently Amended) The apparatus of claim [[2]] 1 wherein the code randomizer produces a help file as an output, the help file specifying how the secret is to be injected into the (n)th executable by the injector, and wherein the injector receives the help file as an input and injects the secret into the (n)th executable according to the help file.

8. (Original) The apparatus of claim 7 wherein the code randomizer produces the help file as an embedded portion of the (n) executable.

9. (Original) The apparatus of claim 1 further comprising a signature generator receiving the (n)th executable as an input, generating a digital signature for the (n)th executable, coupling the generated digital signature to the (n)th executable, and producing the coupled (n)th executable and digital signature as an output, wherein the coupled (n)th executable and digital signature and the encrypted (n)th key file are to be forwarded to the requesting DRM system.

10. (Currently Amended) A method for producing a new ((n)th) black box for a digital rights management (DRM) system, the (n)th black box for being installed in the DRM system and for performing decryption and encryption functions in the DRM system, the (n)th black box being produced and delivered to the DRM system upon request therefrom and including a new ((n)th) executable and a new ((n)th) key file, the (n)th key file having a new ((n)th) set of black box keys and a number of old sets of black box keys, the request including an old ((n-1)th) key file having the old sets of black box keys, the method comprising:

receiving a master executable and randomized optimization parameters;

producing the (n)th executable based on the received master executable and the received randomized optimization parameters and based on a code optimization / randomization technique;

receiving the (n-1)th key file and the (n)th set of black box keys;

extracting the old sets of black box keys from the (n-1)th key file;

producing the (n)th key file including the (n)th set of black box keys and the old sets of black box keys as an output based on the extracted old sets of black box keys from the (n-1)th key file and the received (n)th set of black box keys; and

forwarding the produced (n)th executable and the produced (n)th key file to the requesting DRM system,

<u>wherein producing the (n)th executable comprises producing the (n)th executable with space reserved therein for additional information to be injected by an injector, and</u>

wherein producing the (n)th key file includes encrypting the (n)th set of black box keys and the old sets of black box keys according to a secret, and wherein producing the (n)th executable comprises injecting the secret into at least a portion of the reserved space.

11. (Original) The method of claim 10 wherein the old sets of keys in the (n-1)th key file are encrypted according to a secret of an (n-1)th executable, and wherein extracting the old sets of keys comprises obtaining the secret of the (n-1)th executable and applying the secret to the encrypted old sets of keys in the (n-1)th key file.

12. (Original) The method of claim 11 wherein the request includes the (n-1)th executable, wherein the secret is embedded in the (n-1)th executable, and wherein obtaining the secret of the (n-1)th executable comprises extracting the secret from the (n-1)th executable.

13. (Original) The method of claim 11 wherein the secret is maintained in a database, and wherein extracting the old sets of keys comprises obtaining the secret from the database.

14. (Original) The method of claim 11 wherein the secret is included in the (n-1)th key file, and wherein extracting the old sets of keys comprises obtaining the secret from the (n-1)th key file.

15. (Original) The method of claim 14 wherein the secret is included in the (n-1)th key file encrypted according to a super secret (SUPER(secret)), and wherein extracting the old sets of keys comprises obtaining (SUPER(secret)) from the (n-1)th key file, obtaining the super secret, and applying the super secret to (SUPER(secret)) to obtain the secret.

16. (Original) The method of claim 10 wherein producing the (n)th key file includes encrypting the (n)th set of black box keys and the old sets of black box keys according to a secret.

17. (Original) The method of claim 16 wherein producing the (n)th key file includes encrypting the (n)th set of black box keys and the old sets of black box keys according to a secret derived from the (n)th set of black box keys.

18. (Original) The method of claim 16 wherein producing the (n)th executable comprises embedding the secret therein.

19. (Original) The method of claim 16 further comprising maintaining the secret in a database.

20. (Original) The method of claim 16 wherein producing the (n)th key file further includes placing the secret in the (n)th key file.

21. (Original) The method of claim 20 wherein producing the (n)th key file further includes encrypting the secret according to a super secret (SUPER(secret))and placing (SUPER(secret)) in the (n)th key file.

22. (Original) The method of claim 10 wherein the DRM system resides on a computing device having a hardware ID (HWID) associated therewith, wherein the (n-1)th key file further has the HWID therein, wherein the method further comprises extracting the HWID from the (n-1)th key file, and wherein producing the (n)th key file comprises inserting the extracted HWID into the (n)th key file.

23-24 (Canceled)

25. (Currently Amended) The method of claim [[23]] 10 wherein the DRM system resides on a computing device having a hardware ID (HWID) associated therewith, wherein the (n-1)th key file further has the HWID therein, wherein the method further comprises extracting the HWID from the (n-1)th key file, and wherein producing the (n)th executable comprises injecting the extracted HWID into at least a portion of the reserved space.

26. (Canceled)

27. (Currently Amended) The method of claim [[26]] 10 wherein producing the (n)th key file includes encrypting the (n)th set of black box keys and the old

sets of black box keys according to a secret, and wherein producing the (n)th executable comprises injecting the secret into at least a portion of the reserved space in a manner hidden except to the (n)th executable.

28. (Original) The method of claim 10 wherein the DRM system resides on a computing device having a hardware ID (HWID) associated therewith, wherein the (n-1)th key file further has the HWID therein, wherein the method further comprises extracting the HWID from the (n-1)th key file, and wherein producing the (n)th executable comprises producing the (n)th executable based at least in part on the extracted HWID and based on a code optimization / randomization technique.

29. (Original) The method of claim 10 comprising:

receiving, at a code optimizer / randomizer, a master executable and randomized optimization parameters as inputs;

producing, at the code optimizer / randomizer, the (n)th executable as an output based on the inputs thereto;

receiving, at a key manager, the (n-1)th key file and the (n)th set of black box keys as inputs;

extracting, at the key manager, the old sets of black box keys from the (n-1)th key file;

producing, at the key manager, the (n)th key file including the (n)th set of black box keys and the old sets of black box keys as an output based on the inputs thereto; and

forwarding the produced (n)th executable and the produced (n)th key file to the requesting DRM system.

30. (Currently Amended) A method for producing a new ((n)th) black box for a digital rights management (DRM) system, the (n)th black box for being installed in the DRM system and for performing decryption and encryption functions in the DRM system, the (n)th black box being produced and delivered to the DRM system upon request therefrom and including a new ((n)th) executable, the method comprising:

receiving a master executable and randomized optimization parameters;

producing the (n)th executable based on the received master executable and the received randomized optimization parameters and based on a code optimization / randomization technique; and

forwarding the produced (n)th executable to the requesting DRM system,

wherein producing the (n)th executable comprises producing the (n)th executable with space reserved therein for additional information to be injected by an injector, and

wherein the (n)th black box further includes a new ((n)th) key file , the (n)th key file having a new ((n)th) set of black box keys and a number of old sets of black box keys, wherein the (n)th key file is produced to include the (n)th set of black box keys and the old sets of black box keys encrypted according to a secret, and wherein producing the (n)th executable comprises injecting the secret into at least a portion of the reserved space.

31-32 (Canceled)

33. (Currently Amended) The method of claim [[31]] 30 wherein the DRM system resides on a computing device having a hardware ID (HWID) associated therewith, wherein the request from the DRM system includes the HWID, and wherein producing the (n)th executable comprises injecting the included HWID into at least a portion of the reserved space.

34. (Canceled)

35. (Currently Amended) The method of claim [[34]] 30 wherein producing the (n)th executable comprises injecting the secret into at least a portion of the reserved space in a manner hidden except to the (n)th executable.

36. (Original) The method of claim 30 wherein the DRM system resides on a computing device having a hardware ID (HWID) associated therewith, wherein the request from the DRM system includes the HWID, and wherein producing the (n)th executable comprises producing the (n)th executable based at least in part on the included HWID and based on a code optimization / randomization technique.

37. (Original) The method of claim 30 comprising:

receiving, at a code optimizer / randomizer, a master executable and randomized optimization parameters as inputs; and

producing, at the code optimizer / randomizer, the (n)th executable as an output based on the inputs thereto.

38-50 (Canceled)